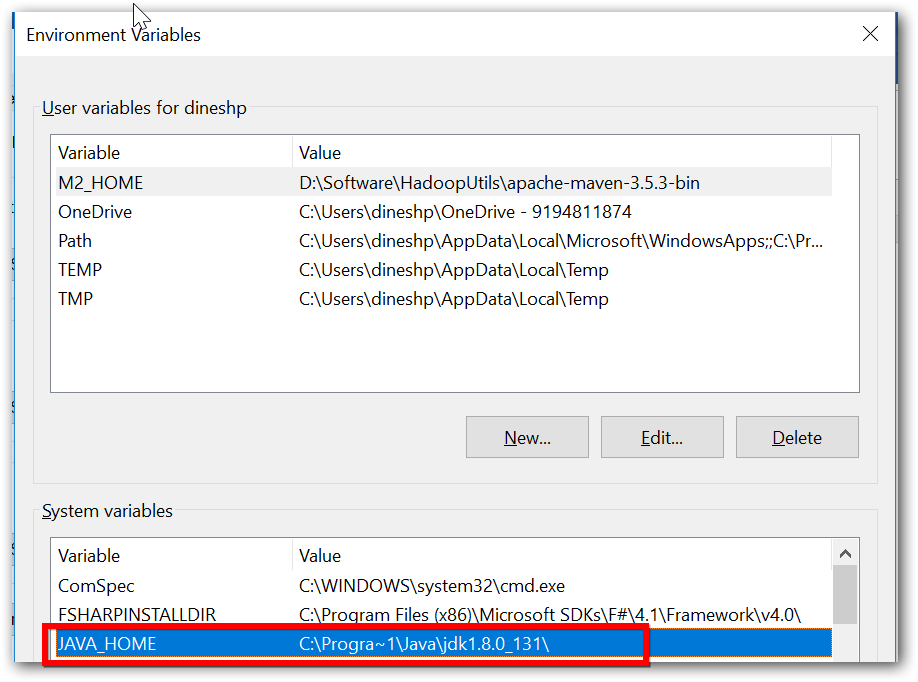
Hands on #1 – Hadoop + Azure Blob cluster

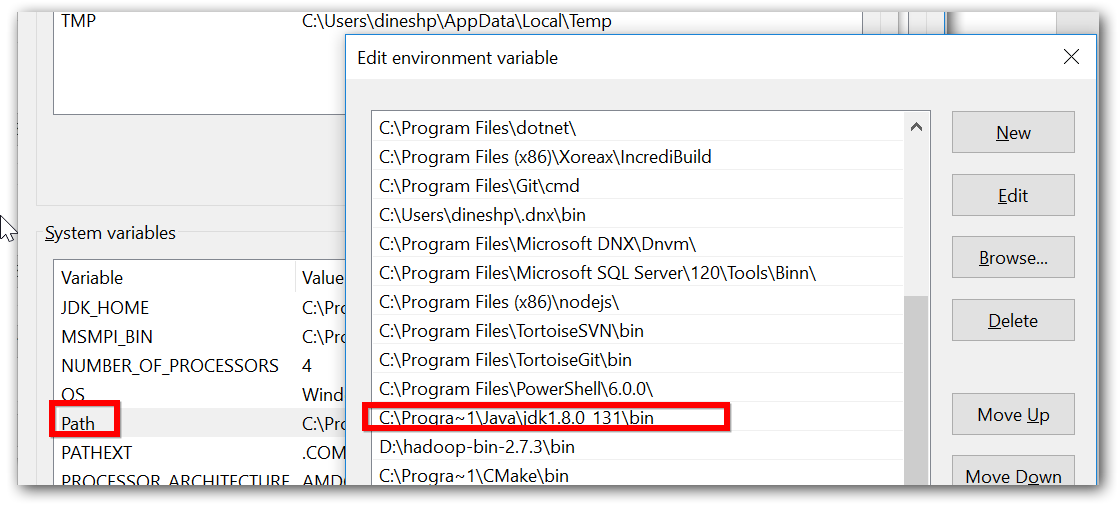
# Pre-requisites

1. Install **7z1801-x64** in your machine.
2. Untar **“hadoop-3.0.1.tar”** using 7z. Prefer any other drive apart from OS drive. For e.g. if ‘C:\’ is OS drive, have Hadoop package in ‘D:\’ or some other drive.
3. Install Java 8 in your machine.
4. Set JAVA\_HOME in environment variables – system variables

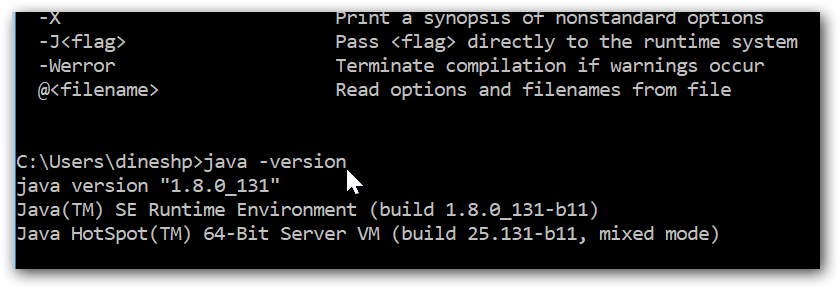
<https://confluence.atlassian.com/doc/setting-the-java_home-variable-in-windows-8895.html>



1. Add Java’s bin location PATH

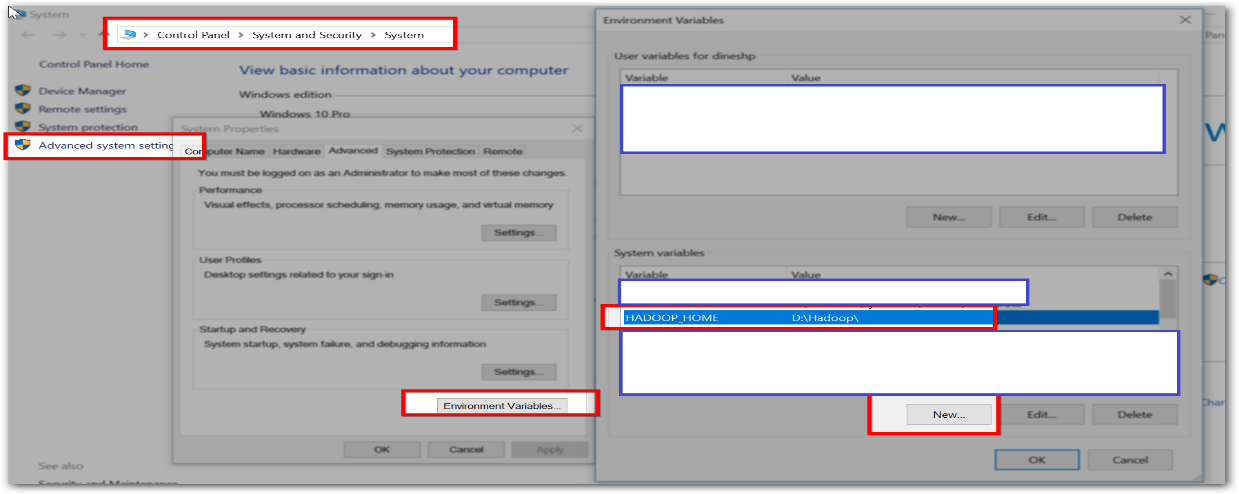


1. Ensure java home and path is set properly.
   1. Open command prompt
   2. Execute ‘javac’ or ‘java -version’ command.



1. Set HADOOP\_HOME as environment variables (without bin),

|  |  |
| --- | --- |
| HADOOP\_HOME | D:\hadoop\3.0.1 |



# Hadoop cluster installation

## Configuration

1. For blob storage you need to copy the following jars from the [**Data** folder](https://drive.google.com/open?id=1IhQvk_KAqBD96_svvP8QtiEIqhowwxWY) to “D:\hadoop-3.0.1\share\hadoop\hdfs\lib”
   1. azure-storage-8.0.0.jar
   2. hadoop-azure-3.0.1.jar
2. Do the following changes in specified file name located in directory “D:\hadoop-3.0.1\etc\hadoop\”

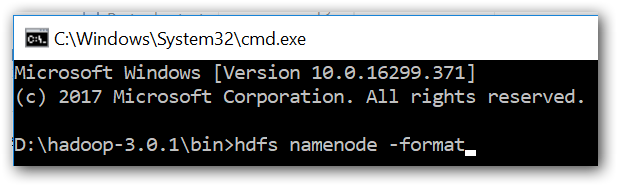
|  |  |  |
| --- | --- | --- |
| **File name** | **Properties** | **Comments** |
| hadoop-env.cmd  (Windows command script file) | set **HADOOP\_PREFIX**=D:\hadoop-3.0.1  set HADOOP\_CONF\_DIR=%HADOOP\_PREFIX%\etc\hadoop  set YARN\_CONF\_DIR=%HADOOP\_CONF\_DIR%  set PATH=%PATH%;%HADOOP\_PREFIX%\bin | Add at the *end* of file, after 90th line.  **HADOOP\_PREFIX** is the Hadoop package unzipped location. |
| core-site.xml | <configuration>  <property>  <name>fs.defaultFS</name>  <value>hdfs://localhost:9000</value>  </property>  <property>  <name>**fs.azure.account.key.dineshazuredemo.blob.core.windows.net**</name>  <value>**xxxxxxxxxxxxxx**</value>  <description>Storage account Access key copied from Azure portal</description>  </property>  <property>  <name>fs.AbstractFileSystem.wasb.impl</name>  <value>org.apache.hadoop.fs.azure.Wasb</value>  </property>  <property>  <name>fs.wasb.impl</name>  <value>org.apache.hadoop.fs.azure.NativeAzureFileSystem</value>  </property>  <property>  <name>fs.hdfs.impl</name>  <value>org.apache.hadoop.hdfs.DistributedFileSystem</value>  </property>  <property>  <name>fs.file.impl</name>  <value>org.apache.hadoop.fs.LocalFileSystem</value>  </property>  </configuration> | Replace the content instead of <configuration></configuration>  Changes in **bold** are required if you need to configure blob storage in the cluster. If not specified, the cluster will be with local storage alone. |
| hdfs-site.xml | <configuration>  <property>  <name>dfs.replication</name>  <value>1</value>  </property>  <property>  <name>dfs.namenode.name.dir</name>  <value>file:///d:/Data/NameNode</value>  </property>  <property>  <name>dfs.datanode.data.dir</name>  <value>file:///d:/Data/DataNode</value>  </property>  </configuration> | Replace the content instead of <configuration></configuration>  Prefer having any other drive apart from ‘C:\’ |

## Format your cluster and make your file system ready for use

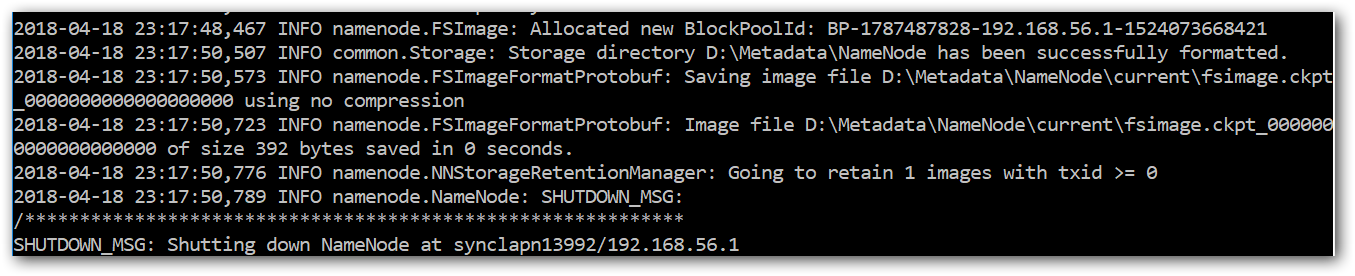
Open command prompt as ‘Administrator’ and navigate to Hadoop bin path,

1. Format the cluster

\bin> hdfs namenode -format



1. Ensure that format got successfully completed with similar message in below image,

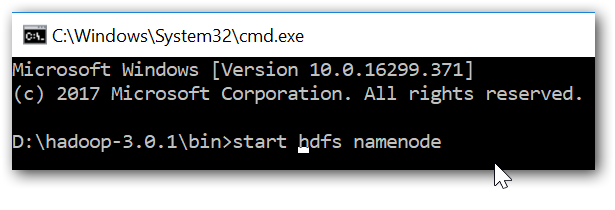


## Start Hadoop cluster services

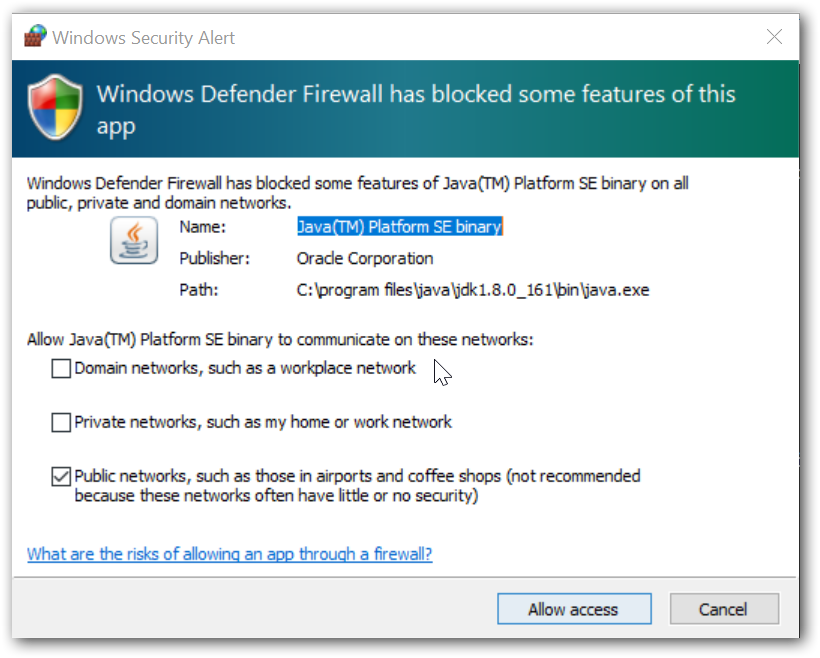
Open command prompt as ‘Administrator’ and navigate to Hadoop bin path,

1. Start NameNode

\bin> start hdfs namenode

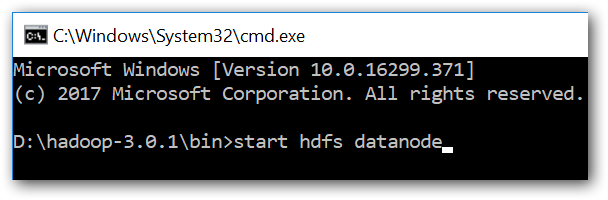


If such a popup is showed while starting service, click ‘Allow access’



1. Start DataNode

\bin> start hdfs datanode



1. Web UI - HDFS - <http://localhost:9870/dfshealth.html#tab-overview>

# Work-out samples

Open command prompt as ‘Administrator’ and navigate to Hadoop bin path,

Do not workout from points 1 to 6. I will do by myself. As its required for 1 time.

1. ~~In the ‘dineshazuredemo’ blob, create a new container called ‘newcontainer’~~
2. ~~Create a folder called ‘Data’ in Blob,~~

**~~bin>~~** ~~hadoop fs -mkdir wasb://newcontainer@dineshazuredemo.blob.core.windows.net/Data~~

1. ~~Upload data into Blob from local file system via Hadoop,~~

**~~bin>~~** ~~hdfs dfs -put "D:\~~ ~~Workshop\Data\word-count-data.txt" wasb://newcontainer@dineshazuredemo.blob.core.windows.net/Data~~

**~~bin>~~** ~~hdfs dfs -put "D:\~~ ~~Workshop\Data\word-count-data.txt" wasb://newcontainer@dineshazuredemo.blob.core.windows.net/Data~~

1. ~~Command to Copy from Azure Blob to Azure Blob,~~

**~~bin>~~** ~~hadoop distcp wasb://newcontainer@dineshazuredemo.blob.core.windows.net/Data wasb://newcontainer@dineshazuredemo.blob.core.windows.net/Data-Copy~~

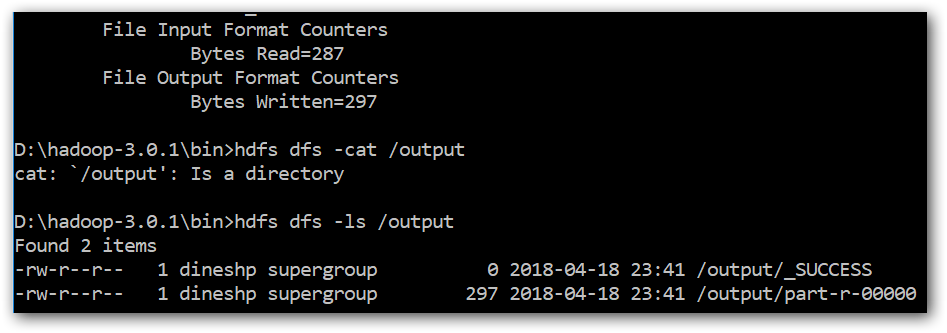
1. ~~Command to Copy from Azure Blob to Hadoop HDFS storage,~~

**~~bin>~~** ~~hadoop distcp wasb://newcontainer@dineshazuredemo.blob.core.windows.net/Data /Data-Copy~~

1. ~~Do Word count for a file present in Azure Blob with Hadoop,~~

~~Replace the ‘hadoop-mapreduce-examples-3.0.1.jar’ file location based on your Hadoop folder,~~

**~~bin>~~** ~~yarn jar "D:\hadoop-3.0.1\share\hadoop\mapreduce\hadoop-mapreduce-examples-3.0.1.jar" wordcount wasb://newcontainer@dineshazuredemo.blob.core.windows.net/Data wasb://newcontainer@dineshazuredemo.blob.core.windows.net/output~~

~~~~

1. Command to list the “/output” directory which has the result stored in file /output/part-r-00000

**bin>** hdfs dfs -ls wasb://newcontainer@dineshazuredemo.blob.core.windows.net/output

1. Command to check the output of word count:

**bin>** hdfs dfs -cat wasb://newcontainer@dineshazuredemo.blob.core.windows.net/output/part-r-00000

